

For: **SYSTEM FOR MANAGING  
DATA IN A DISTRIBUTED  
COMPUTING SYSTEM**

## Date \_\_\_\_\_

## Page 1 of 26

**I. REAL PARTY IN INTEREST**

As evidenced by the assignment recorded at Reel/Frame 015260/0480, the subject application is owned by OpenTV, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 275 Sacramento Street, San Francisco, CA 94111.

**II. RELATED APPEALS AND INTERFERENCES**

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

**III. STATUS OF CLAIMS**

Claims 1-25 are pending and rejected, and are the subject of this appeal. A copy of claims 1-25 as on appeal is included in the Claims Appendix hereto.

**IV. STATUS OF AMENDMEMNTS**

No amendments to the claims have been submitted subsequent to the final rejection.

**V. SUMMARY OF CLAIMED SUBJECT MATTER**

The subject matter of the present claims generally relates to distributed computing systems and, more particularly, to management of data which is delivered to and/or retrieved by an interactive television receiver.

Claim 1 recites a method for managing data in a distributed computing system, said method comprising:

- receiving data comprising a first module which identifies a plurality of modules for use by an application (e.g., page 11, lines 11-13);
- receiving additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules (e.g., page 12, line 28 – page 15, line 21); and
- performing said update (e.g., page 17, lines 7-10).

Claim 9 recites a receiving device comprising:

- receiving circuitry configured to:
- receive data comprising a first module which identifies a plurality of modules for use by an application (e.g., page 11, lines 11-13);
- receive said plurality of modules; and
- receive additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules (e.g., page 12, line 28 – page 15, line 21);
- a processing unit configured to perform said update (e.g., page 17, lines 7-10).

Claim 17 recites a broadcast station comprising:

a server configured to convey data for use by an application (e.g., page 7, lines 14-22);

a transmitter configured to convey data for broadcast (e.g., page 7, lines 14-22);

and

a processing mechanism configured to:

- generate a plurality of modules corresponding to said data (e.g., page 7, line 24 – page 8, line 22);

- generate a first module which identifies said plurality of modules (e.g., page 7, line 24 – page 8, line 22);

- convey said first module and said plurality of modules (e.g., page 7, line 24 – page 8, line 22); and

- determine a change to said plurality of modules is required (e.g., page 10, line 11 – page 11, line 5);

- generate a second module which identifies fewer than all of said plurality of modules and which identifies said change (e.g., page 10, line 11 – page 11, line 5); and

- convey said second module (e.g., page 10, line 11 – page 11, line 5).

Claim 21 recites a computer accessible medium comprising program instructions, said instructions being executable by a processing device to:

- receive data comprising a first module which identifies a plurality of modules for use by an application (e.g., page 11, lines 11-13);

- receive additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules (e.g., page 12, line 28 – page 15, line 21); and

- perform said update (e.g., page 17, lines 7-10).

**VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

1. Claims 1-16 and 21-25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Number 6,038,319 (hereinafter “Chari”) in view of U.S. Patent Number 6,337,951 (hereinafter “Campbell”).
2. Claims 17-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication Number 20020059645 (hereinafter “Soepenbergr”) in view of U.S. Patent Number 6,337,951 (hereinafter “Campbell”).



## VII. ARGUMENT

1. **Claims 1-16 and 21-25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Number 6,038,319 (hereinafter “Chari”) in view of U.S. Patent Number 6,337,951 (hereinafter “Campbell”).**

Appellant submits each of the pending claims recite features neither disclosed nor suggested in the combination of cited references. For example, claim 1 recites:

A method for managing data in a distributed computing system, said method comprising:

receiving data comprising a first module which identifies a plurality of modules for use by an application;

receiving additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules; and

performing said update.

(emphasis added).

In the Office Action dated January 6, 2008, it is suggested that Campbell discloses the above highlighted features at col. 6, lines 46-64. The cited portion of Campbell is reproduced below:

“In response to the user identification and user module data, the host processor sends, over the communications channel, a host origin date for a host module corresponding to the user module. Responsive to receiving

the host origin date for the host module, the user processor compares the host origin date for the host module to the user origin date for the corresponding user module. In response to determining that the host origin date is more recent than the user origin date, the user processor identifies those host module blocks of information, within the host module, having origin dates more recent than corresponding user module blocks of information, within the user module, as updated blocks. The riser processor then downloads to the user memory the updated blocks. Alternatively, in a secondary embodiment, the network may be configured such that, in response to determining that the host origin date is more recent than the user origin date, the user processor downloads the host module and replaces the user module with the downloaded host module.” (Campbell, col. 6, lines 46-64, emphasis added).

In Campbell, the user processor receives a host origin date. However, the host origin date is simply a date and is not an identification of “fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules”. The host origin date is data within a host module and it is compared to data in the user module, such as the user origin date. In Campbell, in the user terminal which receives the host origin date, the user module is identified by the user identification data and not by the host origin date. The host origin date may not have the same value as the user origin date of any user module, much less of the user module identified by the user identification data. Therefore, the host origin date does not identify a user module and does not identify an update to be made to one or more modules. Neither does Chari disclose such features. Accordingly, for at least these reasons, the combination of cited art does not disclose all the features of claim 1 and a prima facie case of obviousness has not been established.

Furthermore, even if one were to accept, for the sake of argument, that the host origin date was equivalent to “additional data that identifies fewer than all of said plurality of modules”, the host origin date does not identify “an update to be made to one or more of said first module and said plurality of modules”. In Campbell, in order to identify “an update to be made,” the user processor must compare the host origin date for the host module to the user origin date for the corresponding user module. Based upon the comparison, some rule or logic is applied in order to conclude whether or not an update is required. However, the host origin date itself does not identify “an update to be made.” Rather, the host origin date simply serves as data input to a process that may or may not result in an identification of an update to be made. Neither does Chari disclose these features. For at least these additional reasons, claim 1 is patently distinct from the combination of cited art.

In addition to the above, claim 1 further recites “...additional data identifies... an update to be made to one or more of said first module and said plurality of modules”. Campbell nowhere discloses additional data identifies an update of said first module “which identifies a plurality of modules.” In Campbell, user identification data identifies a user module. However, the user identification data is not updated by the host origin date or by the host module. Further, Campbell nowhere discloses additional data identifies an update to said plurality of modules. In Campbell, the host module may be used to update a single user module, but the host module is not used to update a plurality of user modules. Neither does Chari disclose such features. For at least these further reasons, claim 1 is believed patently distinguishable from the combination of cited art.

As each of the independent claims 9 and 21 include features similar to claim 1, claims 9 and 21 are patentably distinguished from the cited references alone or in combination for at least reasons similar to those given above.

In addition to the above, the dependent claims recite features not disclosed or suggested by the cited art. For example, the cited art does not disclose the features of claim 2, which recites:

“wherein said first module comprises a main directory module which is pushed, and wherein said additional data comprises a delta directory module which corresponds to said main directory module.” (emphasis added).

In the Final Office Action, it is suggested that Campbell discloses the recited delta directory module in the following:

“After the user terminal has received user identification data and has located a user module in the modules folder, the user processor scans for a user modem. In response to finding a user modem with appropriate settings, the user processor accesses the communications channel to connect with the remotely-located host terminal. The remotely-located host terminal accepts the dialed connection through the host modem. The user module update procedure then continues in a manner equivalent to the user module update procedure followed with the local host.” (Campbell, col. 7, lines 1-9).

However, as discussed above, Campbell merely discloses a host origin date that is data within a host module. Neither the host origin date nor the host module is a delta directory that corresponds to the main directory. In fact, neither one is even a directory. Appellant has reviewed the above portion and the entirety of Campbell and finds no teaching or suggestion of a “delta directory module which corresponds to said main directory module.” Neither does Chari disclose the features of claim 2. Accordingly,

claim 2 is patentably distinct from the cited references alone or in combination for these additional reasons as well.

Still further, as the recited update to be made is not disclosed in the cited art, the features of claims 3 and 5 that relate to said update are not disclosed by the combination of cited art.

Also, as the recited delta directory module is not disclosed in the cited art, the features of claim 4 that relate to the delta directory module are not disclosed by the combination of cited art.

**2. Claims 17-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication Number 20020059645 (hereinafter “Soepenbergl”) in view of U.S. Patent Number 6,337,951 (hereinafter “Campbell”).**

Claim 17 recites, in part

“... a processing mechanism configured to:

generate a plurality of modules corresponding to said data;

generate a first module which identifies said plurality of modules;

convey said first module and said plurality of modules; and

determine a change to said plurality of modules is required;

generate a second module which identifies fewer than all of said plurality

of modules and which identifies said change; and

convey said second module.”

(Emphasis added).

In the Office Action dated January 6, 2008, it is suggested that Campbell discloses the above highlighted features. These rejections rely on the same teachings of Campbell as were cited in the rejection of claim 1. In Campbell, as noted above, the user processor receives a host origin date. However, the host origin date of Campbell is simply a date and is not “a second module which identifies fewer than all of said plurality of modules and which identifies said change,” as is recited in claim 17. The host origin date is data within a host module and it is compared to data in a user module, such as the user origin date. In Campbell, in the user terminal which receives the host origin date, modules are identified by the user identification data and not by the host origin date. The host origin date may not have the same value as the user origin date of any user module, much less of the user module identified by the user identification data. Therefore, the host origin date does not identify “fewer than all of said plurality of modules,” as is recited in claim 17. Also, in Campbell, in order to identify a particular change that is required, the user processor must compare the host origin date for the host module to the user origin date for the corresponding user module. The host origin date does not identify “a change to said plurality of modules is required.” Rather, the host origin date simply serves as data input to a process that may or may not result in a determination that a change to said plurality of modules is required. Nor are these features found in Soepenbergs. For at least these reasons, claim 17 is patently distinct from the combination of cited art. As each of dependent claims 18-20 includes the features of claim 17 on which it depends, each of dependent claims 18-20 is patentably distinct for at least the above reasons.

In addition, regarding claim 17, on page 6 of the Office Action dated January 6, 2008, it is suggested Soepenbergs discloses a server configured to “generate a first module which identifies said plurality of modules,” at [0010] and [0016]. However, the cited portions of Soepenbergs teach:

“Since modules are broadcast in MPEG-2 transport streams, and each module is broadcast in the private data sections of an elementary stream, then typically a large number of modules will share the same elementary stream and a complete object carousel will generally be carried on only a limited number of elementary streams (typically fewer than 5).” [0010]

“In a transmission system, or transmitter or receiver component thereof as recited above, the file and directory modules may be comprised in discrete data portions carried in an elementary data stream, with said predetermined grouping formulation for storage being at the elementary level. Alternately, the file and directory modules may be comprised in discrete data portions carried in an elementary data stream, with said predetermined grouping formulation for storage being at the module level. In either arrangement, the data including file and directory modules may further comprise a version indicator to identify updates, with said modules further comprising discrete data portions carried in an elementary data stream, with said predetermined grouping formulation for storage being at the elementary level.” [0016]

As may be seen from the above, Soepenber merely describes the transmission of file and directory objects. Such file and directory objects correspond to those of a typical file system. Appellant has reviewed the entire disclosure of Soepenber and submits there exists no disclosure of a “first module which identifies a plurality of modules” corresponding to data for use by an application, as recited. In contrast, Soepenber discloses transmitting modules that include files and/or directories. Additionally, the directories described in Soepenber are simply logical containers for files and do not identify a plurality of modules for an application as recited. Further, a module’s version indicator does not identify a plurality of modules used for an application. On page 8, paragraph 7, of the Office Action dated January 10, 2008, in response to Appellant’s

arguments regarding a first module identifying a plurality of modules, the Examiner suggests:

“Soepenbergs disclose an interactive television system enable television sets to be used to provide various new means for providing services to viewers (see [0001]). Such interactive television applications of Soepenbergs consist of one or more programs modules, wherein one module can identifies all of the modules. It is important to note that a carousel is defined as a set of modules, where a module from carousel interacts (identifies) with a plurality of modules of another carousel (see [0035]).”

However, the cited portion of Soepenbergs merely discloses broadcasting modules in a carousel in a particular order. More specifically, Soepenbergs discloses:

“Providing modules on demand, as described above, requires a special interface between the multimedia platform-specific device and the storage device. The multimedia platform-specific device may not have such an interface and it may only expect a (partial) transport stream input. In that case, the storage device has to reconstruct the object carousel. Because the performance of the application may depend on the order in which the modules are being transmitted, and the broadcaster has probably put the modules in a performance-wise optimal order, it is useful to send out the modules with the same relative timestamps as in the original broadcast. The device supports the recording of the (relative) time stamps of transmission of each module that is part of the recording and the usage of these time stamps in the reconstruction of the object carousel. The time stamps can be stored as a list of tuples (time, module, version). This



storage is in addition to the storage of the modules itself (one copy for each version of the module).”

As seen from the above, there is no disclosure of a first module that identifies a plurality of modules as recited. Instead, modules include timestamps through which their versions may be determined. None of the modules identify a plurality of other modules as corresponding to data for use by an application. For at least these additional reasons, claim 17 is patently distinct from the combination of cited art. Claims 18-20 are patentably distinct for at least the reasons given in the discussion of claim 17.

Additionally, on page 8 of the Office Action dated January 10, 2008 regarding claim 18, it is suggested “Chari discloses the claimed ‘main directory module’” (col. 1, lines 60-67). However, claim 18 is rejected on the basis of Soepenbergh and Campbell, not Chari. Neither Soepenbergh nor Campbell disclose such a directory module. Therefore, Appellant submits the Examiner has not shown that claim 18 is unpatentable over Soepenbergh and Campbell and a prima facie case of obviousness has not been established.

**Conclusion**

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-25 was improper, and reversal of the examiner's decision is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Appellant hereby petitions for such an extension. the Commissioner is hereby authorized to charge any fees which may be required to Deposit Account No. 501505/5266-10500/RDR.

Respectfully submitted,

/Rory D. Rankin/

Rory D. Rankin

Reg. No. 47,884

Attorney for Appellants

Meyertons, Hood, Kivlin,  
Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
(512) 853-8850

Date: September 10, 2008

### **VIII. CLAIMS APPENDIX**

The claims on appeal are as follows.

1. (Previously Presented) A method for managing data in a distributed computing system, said method comprising:  
receiving data comprising a first module which identifies a plurality of modules for use by an application;  
receiving additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules; and  
performing said update.
2. (Original) The method of claim 1, wherein said first module comprises a main directory module which is pushed, and wherein said additional data comprises a delta directory module which corresponds to said main directory module.
3. (Original) The method of claim 2, wherein said received main directory module is stored in a memory of a receiving device, and wherein said update comprises directly modifying said main directory module stored in memory.
4. (Original) The method of claim 3, wherein said delta directory includes an identifier which indicates a version to which said main directory is updated in response to performing said update identified by said delta directory.
5. (Original) The method of claim 4, wherein in response to performing said update, corresponding changes are effected in one or more of said plurality of

modules, said corresponding changes being selected from the group consisting of: the addition of a new module; the removal of an existing module; and revisions to an existing module.

6. (Original) The method of claim 5, wherein said plurality of modules comprise one or both of executable application code or data for access by said application during execution.
7. (Original) The method of claim 1, wherein said first module is pushed and wherein said method further comprises said application utilizing said additional data to register changes to the first module.
8. (Original) The method of claim 7, wherein said additional data is pulled by said application.
9. (Previously Presented) A receiving device comprising:
  - receiving circuitry configured to:
    - receive data comprising a first module which identifies a plurality of modules for use by an application;
    - receive said plurality of modules; and
    - receive additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules;
  - a processing unit configured to perform said update.
10. (Original) The receiving device of claim 9, wherein said first module comprises a main directory module which is pushed, and wherein said

additional data comprises a delta directory module which corresponds to said main directory module.

11. (Original) The receiving device of claim 10, wherein said processing unit is configured to update said directory module by directly modifying said main directory module.
12. (Original) The receiving device of claim 11, wherein said delta directory includes an identifier which indicates a version to which said main directory is updated in response to performing said update identified by said delta directory.
13. (Original) The receiving device of claim 12, wherein in response to performing said update on the directory module, said processing unit is configured to effect corresponding changes to said plurality of modules, said changes being selected from the group consisting of: the addition of a new module; the removal of an existing module; and revisions to an existing module.
14. (Original) The receiving device of claim 12, wherein said plurality of modules comprise one or both of executable application code or data for access by said application during execution.
15. (Original) The receiving device of claim 9, wherein said first module and said plurality of modules are pushed to said receiving device by a broadcaster, and wherein said additional data is pulled by said application from an alternate source.

16. (Original) The receiving device of claim 9, wherein said first module is pushed to said receiving device, and wherein said application is not permitted to directly access said first module and is configured to utilize said additional data to register changes to the first module.
17. (Original) A broadcast station comprising:
  - an server configured to convey data for use by an application;
  - a transmitter configured to convey data for broadcast; and
  - a processing mechanism configured to:
    - generate a plurality of modules corresponding to said data;
    - generate a first module which identifies said plurality of modules;
    - convey said first module and said plurality of modules; and
    - determine a change to said plurality of modules is required;
    - generate a second module which identifies fewer than all of said plurality of modules and which identifies said change; and
    - convey said second module.
18. (Original) The broadcast station of claim 17, wherein said first module comprises a main directory module, and wherein said additional data comprises a delta directory module.
19. (Original) The broadcast station of claim 18, wherein said processing mechanism is further configured to:
  - generate an updated main directory module which reflects said change; and
  - convey said updated main directory subsequent to conveying said delta directory module.

20. (Original) The broadcast station of claim 17, wherein said processing mechanism comprises executable program instructions executed by a processor.
21. (Previously Presented) A computer accessible medium comprising program instructions, said instructions being executable by a processing device to:
  - receive data comprising a first module which identifies a plurality of modules for use by an application;
  - receive additional data corresponding to said application, wherein said additional data identifies fewer than all of said plurality of modules and identifies an update to be made to one or more of said first module and said plurality of modules; and
  - perform said update.
22. (Original) The computer accessible medium of claim 21, wherein said first module comprises a main directory module which is pushed, and wherein said additional data comprises a delta directory module which corresponds to said main directory module, wherein the received main directory module is stored in a memory of a receiving device and said update comprises directly modifying said main directory module stored in memory.
23. (Original) The computer accessible medium of claim 22, wherein in response to performing said update, corresponding changes are effected in one or more of said plurality of modules, said corresponding changes being selected from the group consisting of: the addition of a new module; the removal of an existing module; and revisions to an existing module.

24. (Original) The computer accessible medium of claim 21, wherein said first module is pushed and wherein said application utilizes said additional data to register changes to the first module.
25. (Original) The computer accessible medium of claim 24, wherein said additional data is pulled by said application.



**IX. EVIDENCE APPENDIX**

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

**X.     RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.